Logos2 MIPI D-PHY Application Guide

(AN04012 , V1.0)

(2021.10.28)

Shenzhen Ziguang Tongchuang Electronics Co., Ltd.

Document version revision history

DATE	DOCUMENT VERSION	REVISION HISTORY
V1.0	2021-10-28	initial version.

Explanation of terminology

ABBREVIATIONS	FULL SPELLING
MIPI	Mobile Industry Processor Interface
DSI	Display Serial Interface

Table of contents

List of figures

List of tables

1. Overview

1.1. Introduction

This document mainly introduces the MIPI D-PHY reference IP products and application documents of Unisoc Tongchuang Electronics Co., Ltd.'s FPGA/CPLD products. This article mainly introduces the function list, interface definition, interface timing, supporting devices and reference designs of MIPI D-PHY.

MIPI D-PHY reference IP implements the D-PHY layer function of the MIPI communication protocol and is mainly used in the Camera Serial Interface (CSI-2) protocol and Display Serial Interface (DSI) protocol.

1.2. Main features

The main functions supported by MIPI D-PHY are as follows:

- Compatible with MIPI Alliance standard D-PHY V1.0 protocol
- Support 1 clock channel, up to 4 data channels
- Input supports dynamic iodelay adjustment
- Output supports dynamic iodelay adjustment
- Data lines support high-speed mode and asynchronous mode (Escape mode)
- Support CSI2/DSI protocol

1.3. Design information

Table 1 Design information

MIPI D-PHY reference design	
Supported devices	Logos2
Support user interface	Parallel user interface
Design files provided	
Designing Documents	Encrypt files
Reference design	Verilog file
Simulation file	Verilog file
Constraint file	fdc file
Development tool support	
Design Tools	PDS Development Kit: Pango Design Suite 2021.1-SP6.5-Beta1 version Synthesis Tool : ADS

1.4. Resource usage

Table 2 Introduction to resource usage

Device Family	Configuration	LUT	FF	DRM	PLL
PG2L100H-6FBG676	D-PHY RX (Including byte alignment, channel alignment, 4LANE)	240	240	0	0
PG2L100H-6FBG676	D-PHY Tx	3	0	0	0

Note: The above resource list uses PDS 2021.1-SP6.5-Beta1 is compiled on PG2L100H .

2. MIPI D-PHY design

MIPI D-PHY includes two subsystem modules: mipi_dphy_tx and mipi_dphy_rx. mipi_dphy_tx can support CSI2/DSI sending (Master) application; mipi_dphy_rx can support CSI2/DSI receiving (Slave) application.

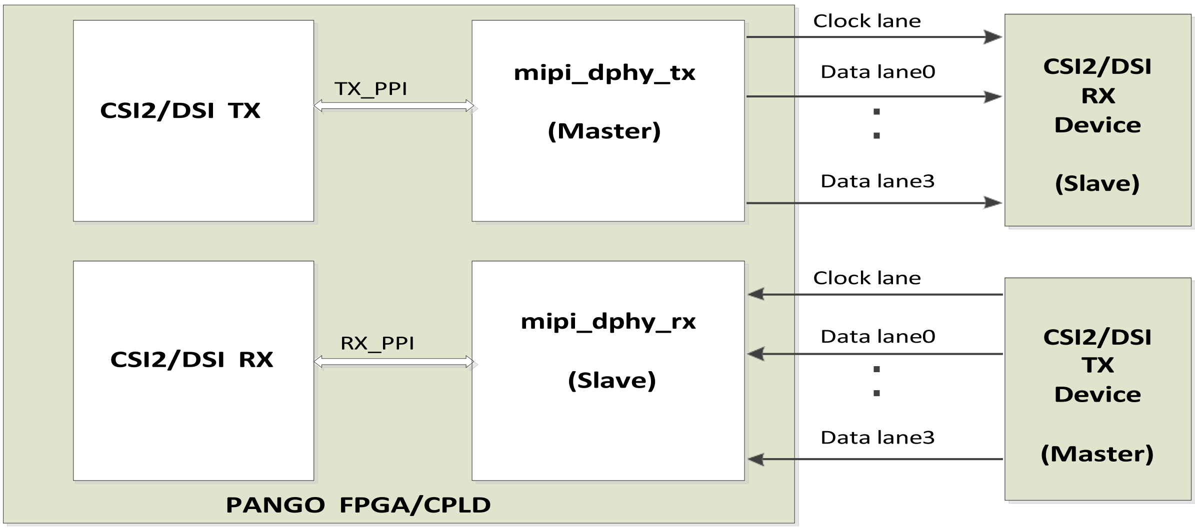


FIGURE 1 MIPI D_PHY APPLICATION INTRODUCTION

2.1 MIPI D-PHY TX (Master) interface

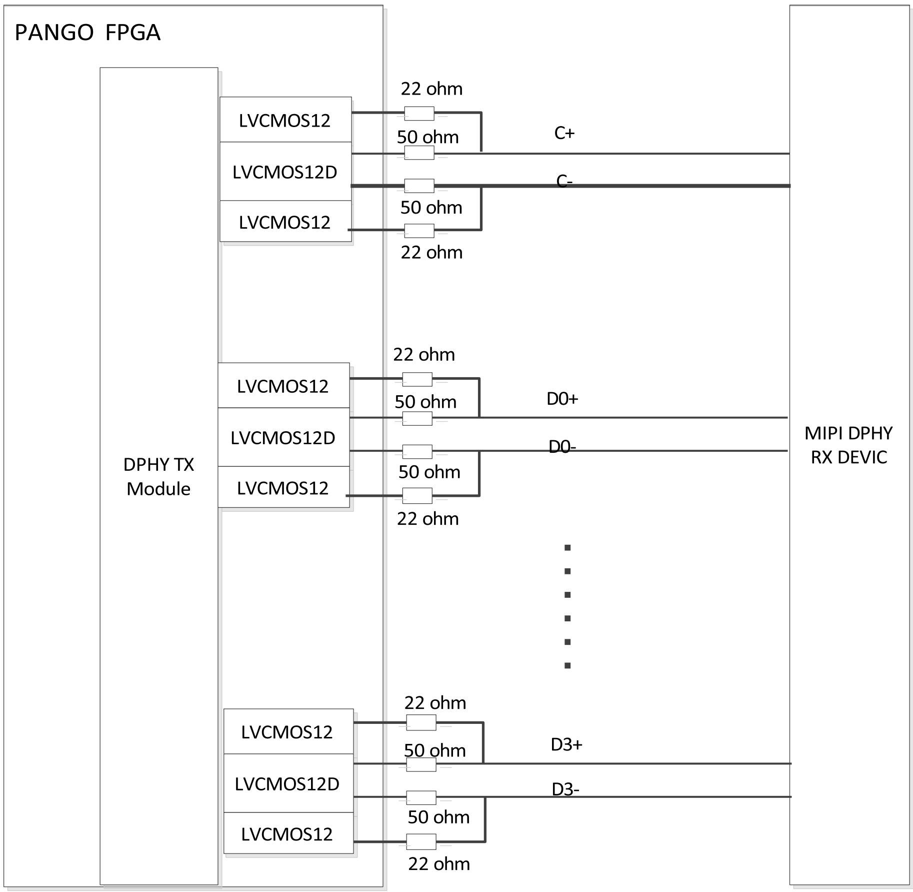


FIGURE 2 MIPI D-PHY TX INTERFACE DIFFERENTIAL ANALOG MIPI SOLUTION

As shown in Figure 2, the transmission is implemented using 4-wire simulation, the high-speed uses the LVCMOS12D level standard, and the low-speed uses the LVCMOS12 level standard.

Table 3 MIPI D_PHY TX interface list

SIGNAL NAME	I/O	BIT WIDTH	DESCRIPTION
INTERNAL USER INTERFACE			
tx_iol_rst	I	1	OSERDES reset signal, active high
tx_div_rst_n	I	1	Divided clock reset signal, active low
tx_hs_i_clk	I	1	Data link high speed serial clock
tx_hs_q_clk	I	1	Clock link high-speed serial clock, same as tx_hs_i_clk frequency etc., there is a 90 ° phase difference
delay_clk_step	I	8	Clock channel delay value, value 0~127 , each step delay 5ps , effective when parameter C_IODELAY_EN is "TRUE"
delay_data_step	I	8~32	Data channel delay value, value 0~127 , each step delay 5ps , effective when parameter D_IODELAY_EN is "TRUE" , each channel occupies 8 bits , and the delay value is configured independently
tx_lp_clk_p	I	1	The clock channel sends data at low speed, p terminal

tx_lp_clk_n	I	1	The clock channel sends data at low speed, n terminal
tx_hs_c_flag	I	1	Clock channel mode, 0 : low speed mode, 1 : high speed mode
tx_lp_dir	I	1~4	Data channel low-speed signal direction 1 : rx , 0 : tx
tx_lp_do_p	I	1~4	Low-speed signal from internal logic output to data port, p-terminal
tx_lp_do_n	I	1~4	Low-speed signal from internal logic output to data port, n-terminal
tx_lp_di_p	O	1~4	The low-speed data signal input from the port to the internal, p-terminal
tx_lp_di_n	O	1~4	Low-speed data signal input from port to internal, n-terminal
tx_hs_d_flag	I	1~4	Data channel mode, 0 : low speed mode, 1 : high speed mode
tx_hs_data	I	8~32	High-speed parallel data
tx_dat_byte_clk	O	1	Data link clock signal, parallel clock
TX PORT MIPI SIGNAL			
tx_lp_io_p	I/O	1~4	Data channel low-speed port, p-terminal
tx_lp_io_n	I/O	1~4	Data channel low-speed port, n-terminal
tx_lp_cp	O	1	Clock channel low-speed port, p - terminal
tx_lp_cn	O	1	Clock channel low-speed port, n - terminal
tx_data_p	O	1~4	Data channel high-speed port, p - terminal
tx_data_n	O	1~4	Data channel high-speed port, n -terminal
tx_clk_p	O	1	High-speed clock port, p - terminal
tx_clk_n	O	1	High-speed clock port, n- terminal

2.2 MIPI D-PHY RX (Slave) interface

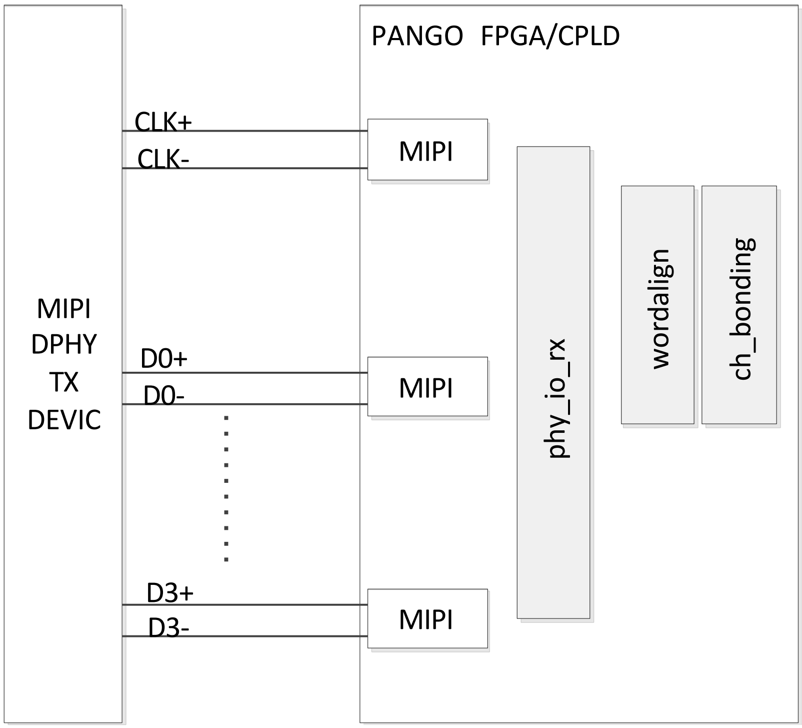


FIGURE 3 MIPI D-PHY RX INTERFACE BLOCK DIAGRAM

MIPI D-PHY RX module mainly consists of phy_io_rx module, wordalign module, and ch_bonding module. Users can use the phy_io_rx module alone, or use the complete MIPI D-PHY RX module.

The phy_io_rx module implements the deserialization of high-speed data in MIPI D-PHY (converting the HS high-speed serial data of the FPGA/CPLD interface into the HS parallel data within the FPGA/CPLD), as well as the control of low-speed data.

The wordalign module implements byte alignment.

The ch_bonding module implements channel alignment.

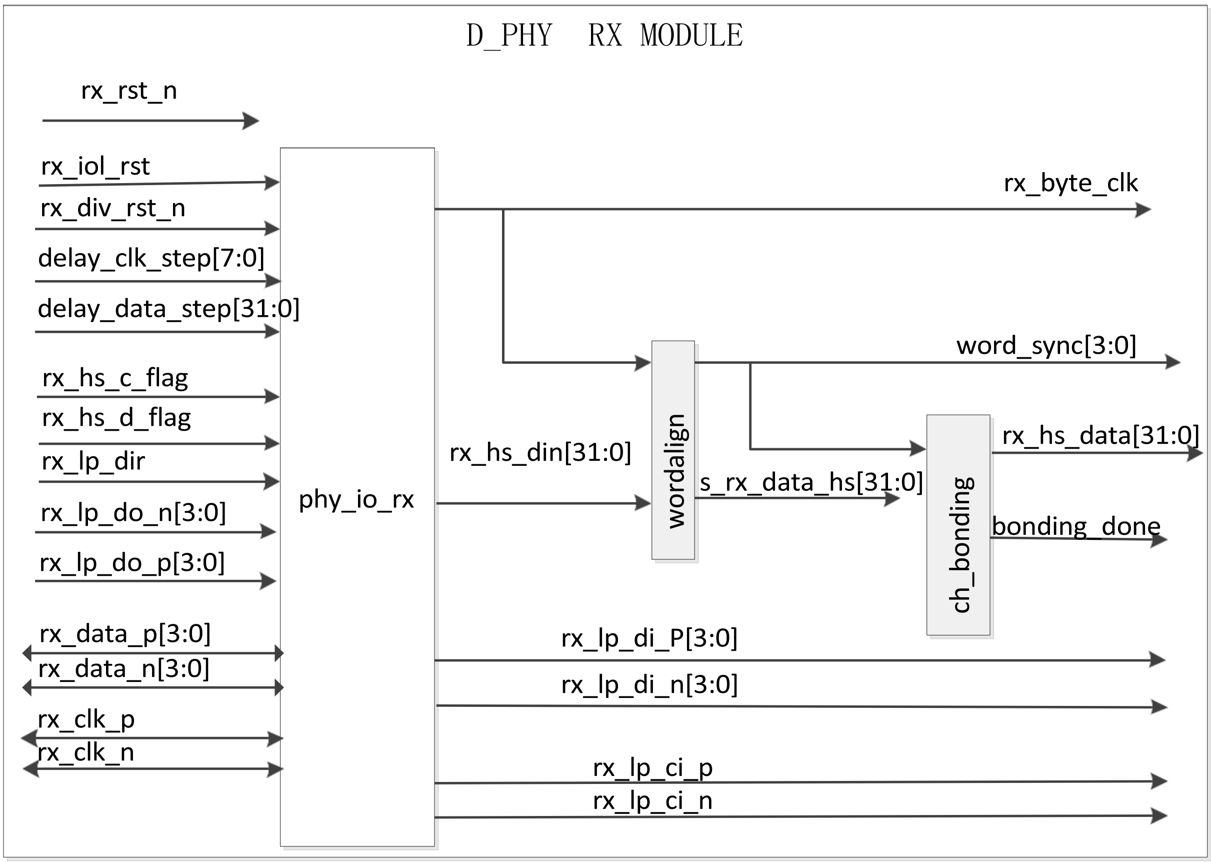


FIGURE 4 MIPI D-PHY RX FUNCTIONAL BLOCK DIAGRAM

Table 4 MIPI D-PHY RX interface list

SIGNAL NAME	I/O	BIT WIDTH	DESCRIPTION
INTERNAL USER INTERFACE			
rx_div_rst_n	I	1	Receives serial clock and divided clock reset input signal, active low
rx_iol_rst	I	1	ISERDES reset signal, active high
rx_rst_n	I	1	Receive module logic reset signal, active low
delay_clk_step	I	8	Clock channel delay value, value 0~247, each step delay is 10ps
delay_data_step	I	8~32	Data channel delay value, ranging from 0 to 247, each step has a delay of 10ps, each channel occupies 8 bits, and the delay value is configured independently
rx_hs_c_flag	I	1	Clock channel mode, 0 : low speed mode, 1 : high speed mode
rx_hs_d_flag	I	1	Data channel mode, 0 : low speed mode, 1 : high speed mode
rx_lp_dir	I	1~4	Data channel low-speed signal direction 1 : rx , 0 : tx
rx_lp_do_p	I	1~4	Low-speed signal output to the data channel port, p - terminal
rx_lp_do_n	I	1~4	Low-speed signal output to data channel port, n - terminal
rx_lp_di_p	O	1~4	Data channel low-speed signal input from port, p

			terminal
rx_lp_di_n	O	1~4	Data channel low-speed signal input from port, n-terminal
rx_lp_ci_p	O	1	Clock channel low-speed data input from port, p terminal
rx_lp_ci_n	O	1	Clock channel low-speed data input from port, n terminal
rx_hs_data	O	8~32	High-speed deserialization of data
rx_byte_clk	O	1	Parallel clock, rx_div_rst_n must be released to generate
word_sync	O	1~4	Byte alignment (8'hb8) identification signal, active high, one parallel clock width
bonding_done	O	1	Channel alignment identification signal, active high, one parallel clock width
RX PORT MIPI SIGNAL			
rx_data_p	I/O	1~4	Data port, p- terminal
rx_data_n	I/O	1~4	Data port, n- terminal
rx_clk_p	I/O	1	The clock port, p -side, must come from the GMCLK pin
rx_clk_n	I/O	1	The clock port, n -terminal, must come from the GMCLK pin

2.3 Parameter definition

Table 5 MIPI D-PHY reference definition

PARAMETER	DESCRIPTION
MIPI D-PHY RX	
M_WORD_ALIGN	Byte alignment function is enabled; Configurable values: "TRUE", "FALSE"; default = "TRUE"
M_CH_BONDING	Channel alignment function enabled; Configurable values: "TRUE", "FALSE" default = "TRUE"
LANE_WIDTH	DPHY Number of Lanes ; supports 1/2/3/4; default = 4
DATA_WIDTH	Parallel data bit width, fixed at 8 default = 8
BUS_WIDTH	Internal data bus width = DATA_WIDTH * LANE_WIDTH default = 32
MIPI D-PHY TX	
C_IODELAY_EN	Clock channel iodelay function enable "TRUE ": Enable, the delay value is approximately equal to 0.6ns + 5ps*delay_clk_step "FALSE " Disable the sending clock channel iodelay function default： "TRUE"
D_IODELAY_EN	Data channel iodelay function enable "TRUE ": Enable, the delay value is approximately equal to 0.6ns + 5ps*delay_data_step "FALSE " Disable the sending data channel iodelay function

	default："TRUE"
CLK_PATTERN	Clock channel high-speed data, value 8'h55 or 8'haa default:8'h55
LANE_WIDTH	Same as above
DATA_WIDTH	Same as above
BUS_WIDTH	Same as above

2.4 Interface timing

Transmit reset timing: first release tx_iol_rst (reset of OSERDES), then release the reset of tx_div_rst_n (reset of high-speed frequency division clock), and finally release tx_gate_en (gate enable of PLL output) to ensure that all OSERDES work at the same time.



FIGURE 5 D-PHY TX RESET TIMING

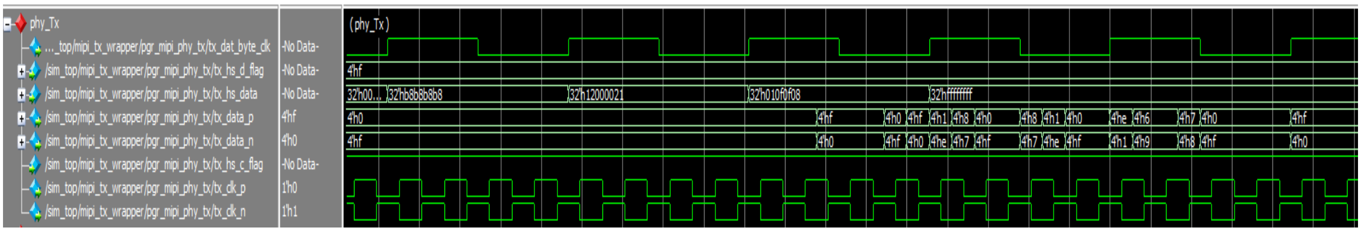


FIGURE 6 D-PHY TX INTERFACE TIMING

Receiving reset timing: first release rx_iol_rst (reset of ISERDES), then release rx_div_rst_n (reset of high-speed frequency division clock) to ensure that all ISERDES work at the same time, and finally release rx_rstn (reset of logic).

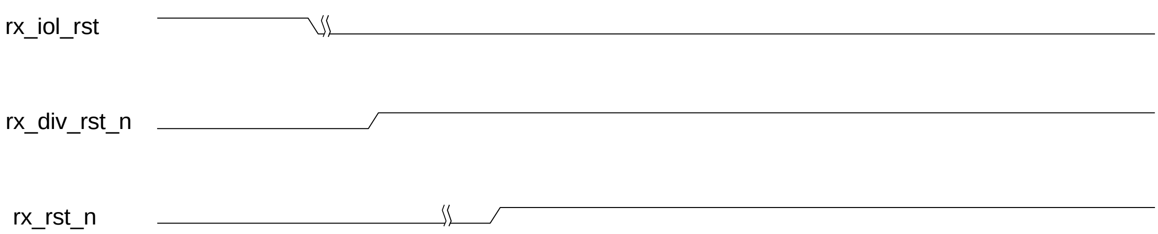


FIGURE 7 D-PHY RX RESET TIMING

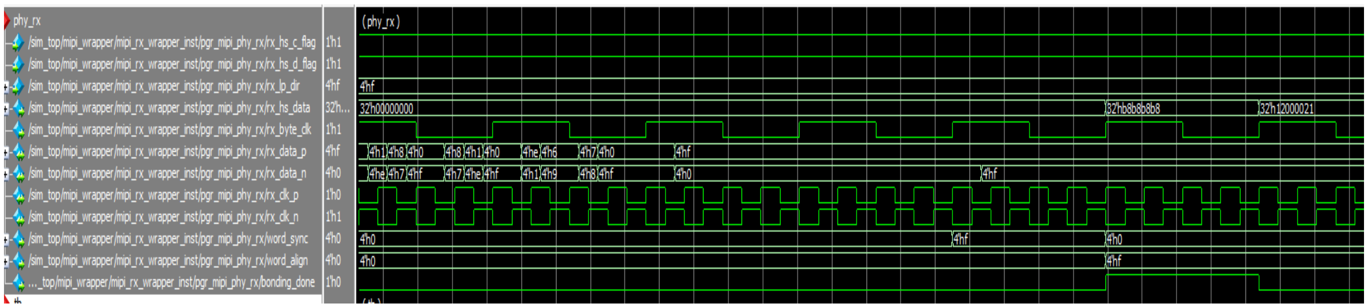


FIGURE 8 D-PHY RX INTERFACE TIMING

3. Reference design

3.1 Functional reference design

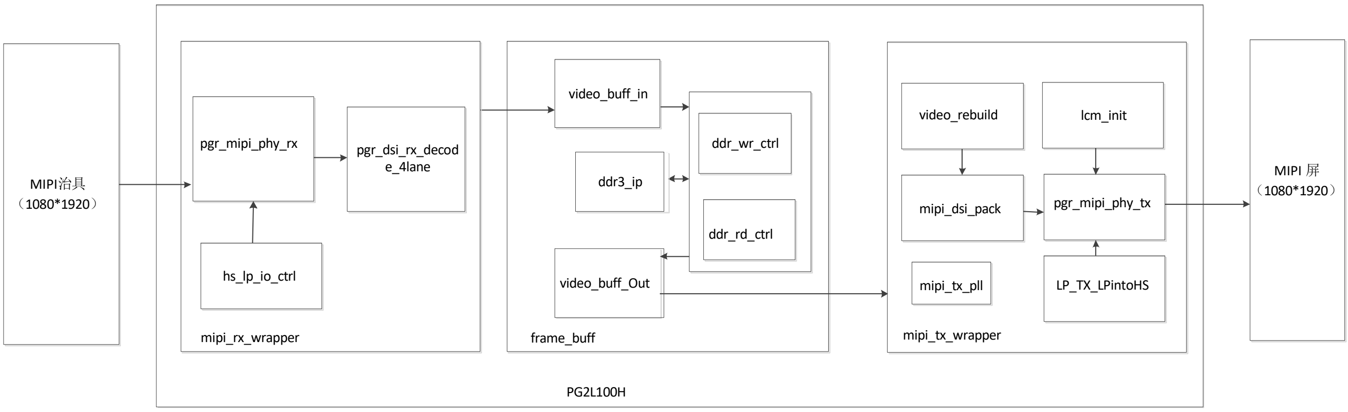


FIGURE 9 REFERENCE DESIGN MODULE STRUCTURE

As shown in Figure 9, the main function implemented by the reference design is to buffer the 1080*1920 MIPI video source through DDR and display it through the MIPI screen.

mipi_rx_wrapper module: MIPI RX top-level module;

pgr_mipi_phy_rx module: MIPI D-PHY RX bottom module, which implements serial-to-parallel conversion, byte alignment and channel alignment;

hs_lp_io_ctrl module: realizes high and low speed mode switching;

pgr_dsi_rx_decode_4lane module: implements decoding of MIPI video streams;

frame_buff module: frame buffer top-level module;

video_buff_in module: implements line caching of input video sources;

ddr3_ip module: ddr controller module;

ddr_wr_ctrl module: implements ddr write operation control;

ddr_rd_ctrl module: implements ddr read operation control;

video_buff_out module: implements line buffering of output videos;

mipi_tx_wrapper module: MIPI TX top-level module;

video_rebuild module: generates the video interface timing required by the mipi_dsi_pack module;

mipi_dsi_pack module: Packaging processing of video data;

LP_TX_LPintoHS module: sends the low-speed signal of the clock channel from low speed to high speed;

lcm_init module: implements the initialization operation of MIPI screen;

mipi_tx_pll module: Provides the high-speed clock and low-speed initialization clock of the mipi_tx_wrapper module;

pgr_mipi_phy_tx module: MIPI D-PHY TX bottom module, which implements parallel-to-serial conversion and low-speed signal transmission;

3.2 Reference Design Interface List

Table 6 Top-level interface definition

SIGNAL NAME	I/O	BIT WIDTH	DESCRIPTION
CLOCK RESET INTERFACE			
rstn_i	input	1	System reset signal, active low
key0_in_n	input	1	Debug button, reserved
sys_clk_i	input	1	Input reference clock, 50MHz
debug	output	4	reserve
Led1	output	1	reserve
bl_ctrl	output	1	reserve
MIPI_FO_RST	output	1	reserve
MIPI PORT			
rx_data_p	inout	4	Receive data port, p end
rx_data_n	inout	4	Receive data port, n end
rx_clk_p	inout	1	Receive clock port, p terminal
rx_clk_n	inout	1	Receive clock port, n terminal
tx_lp_io_p	inout	4	Send data channel low-speed port, p terminal
tx_lp_io_n	inout	4	Send data channel low-speed port, n terminal
tx_lp_cp	output	1	Transmit clock channel low-speed port, p-terminal
tx_lp_cn	output	1	Transmit clock channel low-speed port, n-terminal
tx_data_p	output	4	Send high-speed data port, p-terminal
tx_data_n	output	4	Send high-speed data port, n-terminal
tx_data_p	output	1	Send high-speed clock port, p-terminal
tx_data_n	output	1	Transmit high-speed clock port, n-terminal

Note: DDR3 Interface reference DDR3 Reference design

3.3 Parameter definition

Table 7 Parameter definition

PARAMETER	DESCRIPTION
FRAME_W	The number of effective row pixels of the video source Default: 1080
FRAME_H	The number of valid lines in a frame Default: 1920
M_WORD_ALIGN	Byte alignment enable, optional "TRUE", "FALSE" Default: "TRUE"
M_CH_BONDING	Channel alignment enable, optional "TRUE", "FALSE" Default: "TRUE"
LANE_WIDTH	Number of receiving channels Default value: 4
DATA_WIDTH	Receive channel data bit width Default value: 8
BUS_WIDTH	Receive data bus width, equal to LANE_WIDTH* DATA_WIDTH Default value: 32
DECODE_MODE	Optional "COMMAND_MODE" , " VIDEO_MODE" "VIDEO_MODE ": parse the video source of Video Mode "COMMAND_MODE ": parse Command Mode 's video source Default value: " VIDEO_MODE "
V_SYNC	DECODE_MODE is "VIDEO_MODE" mode, indicating the width of frame synchronization, the unit is line, for example, 3 , indicating that the pulse width of frame synchronization is 3 line widths Default value: 3
H_SYNC	DECODE_MODE is "VIDEO_MODE" mode, indicating the width of horizontal synchronization, the unit is the number of clocks, such as 3 , indicating that the pulse width of horizontal synchronization is 3 clock widths Default value: 3
TX_H_TOTAL	The total number of clocks in the horizontal direction of the transmission timing Default value: 1225
TX_H_ACTIVE	The number of effective pixels in the horizontal direction of the sending timing Default: 1080
TX_H_FP	Send timing horizontal direction leading edge Default: 80
TX_H_WIDTH	Transmission timing horizontal direction horizontal synchronization pulse width Default value: 20
TX_V_TOTAL	Total number of lines in vertical direction of sending timing Default value: 2020
TX_V_ACTIVE	Send timing vertical direction effective line Default value: 1920
TX_V_FP	Send timing vertical direction leading edge Default value: 30
TX_V_WIDTH	Transmit timing vertical synchronization pulse width Default value: 16

3.4 Reference Design Interface Timing

As shown in Figure 10, the interface timing of the mipi_rx_wrapper module when the video source is Video Mode, vsync is high indicating the start of frame synchronization, and the pulse width is determined by the parameter V_SYNC. hsync is high indicating the start of horizontal synchronization, and the pulse width is determined by the parameter H_SYNC.

hs_data is parallel high-speed video data, valid when hs_data_en is high.

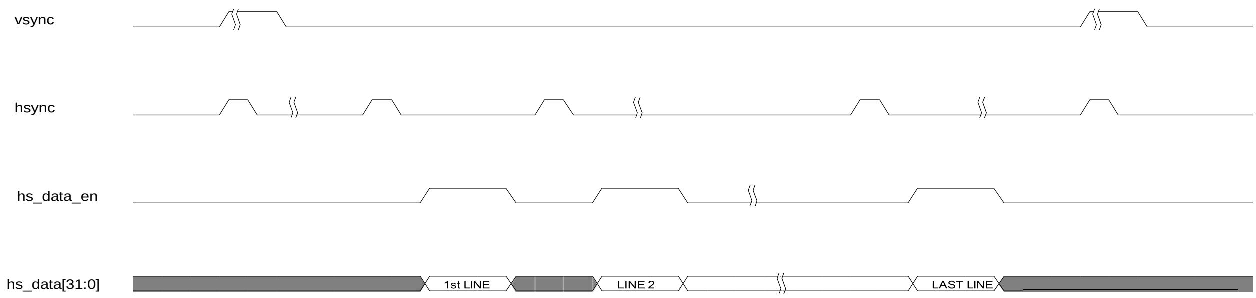


FIGURE 10 DECODING MODULE -VIDEO MODE INTERFACE TIMING

As shown in Figure 11, the interface timing of the mipi_rx_wrapper module when the video source is Command Mode, vsync is high to indicate the start of a frame, 3 clock widths, aligned with the rising edge of hs_data_en in the first line. hs_data is parallel high-speed video data, valid when hs_data_en is high.

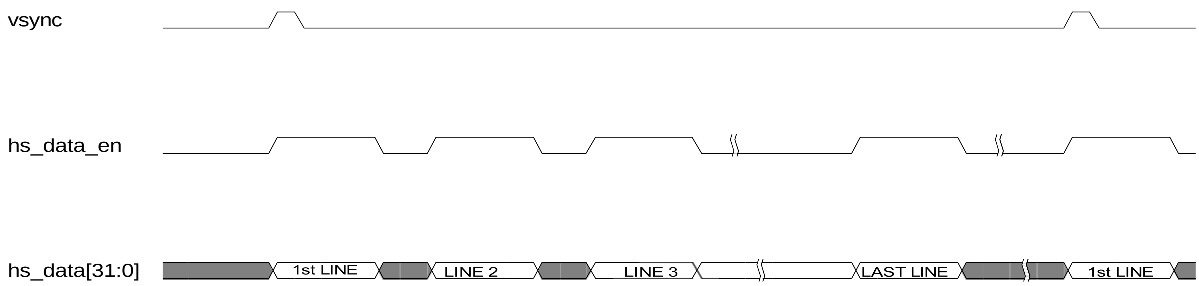


FIGURE 11 DECODING MODULE -COMMAND MODE INTERFACE TIMING

As shown in Figure 12, it is the interface timing of mipi_tx_wrapper module, frame_sync_o is the frame synchronization signal with 3 clock widths, hsync_start_o is the line synchronization signal with 1 clock width, hs_data_req_o is the read data request signal, and the user needs to prepare the next beat of hs_data_req_o for the video data tx_hs_data_i that needs to be sent.

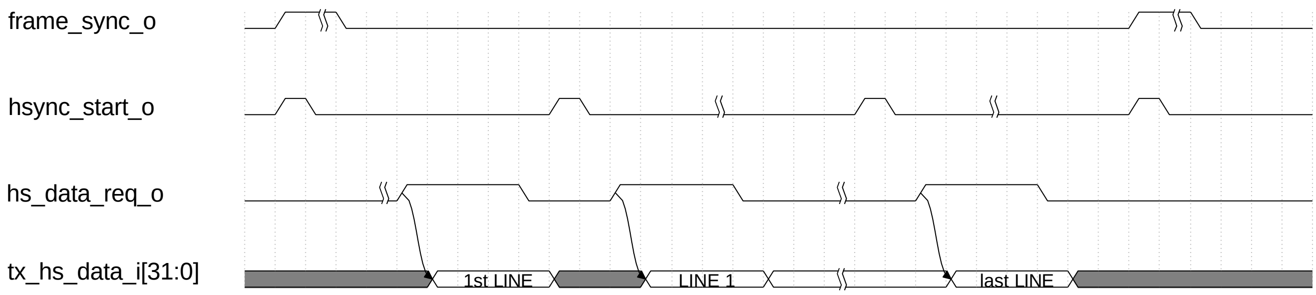


FIGURE 12 MIPI TRANSMIT MODULE INTERFACE TIMING

3.5 Reference design file directory

```
Design example directory structure diagram:
pgr_Logos2_MIPI_DPHY_eval
│
├bench // Simulation test bench
├doc //Design document
├ip //Relevant IP called by design
├pnr // Exampl project directory
│ ├generate_bitstream //Bitstream file generated by PDS
│ ├ipcore //IP CORE file generated by PDS
│ ├mipi_wrapper.pds //PDS project file
│ ├p04w100as01_a0.fdc // Example engineering constraint reference file
│ ├mipi_wrapper_syn.fic //Debug Core debugging file
├simulation //Simulation project directory
│ ├sim_ref //Simulation encrypted file
│ ├sim_file_list.f //Design file list
│ ├sim.bat //Simulation script
│ ├sim.tcl //Simulation TCL script
│ └wave.do //Simulation waveform script
├src //RTL file included in the design example
```

FIGURE 13 FILE DIRECTORY

3.6 Reference design simulation

When performing simulation, the user first sets the installation path of the simulation library in the sim.tcl file and lines 202~225 in the sim_file_list.f file according to the version of PDS installed. After setting the simulation environment, run the sim.bat script. Can. The simulation waveform is shown in Figure 14:

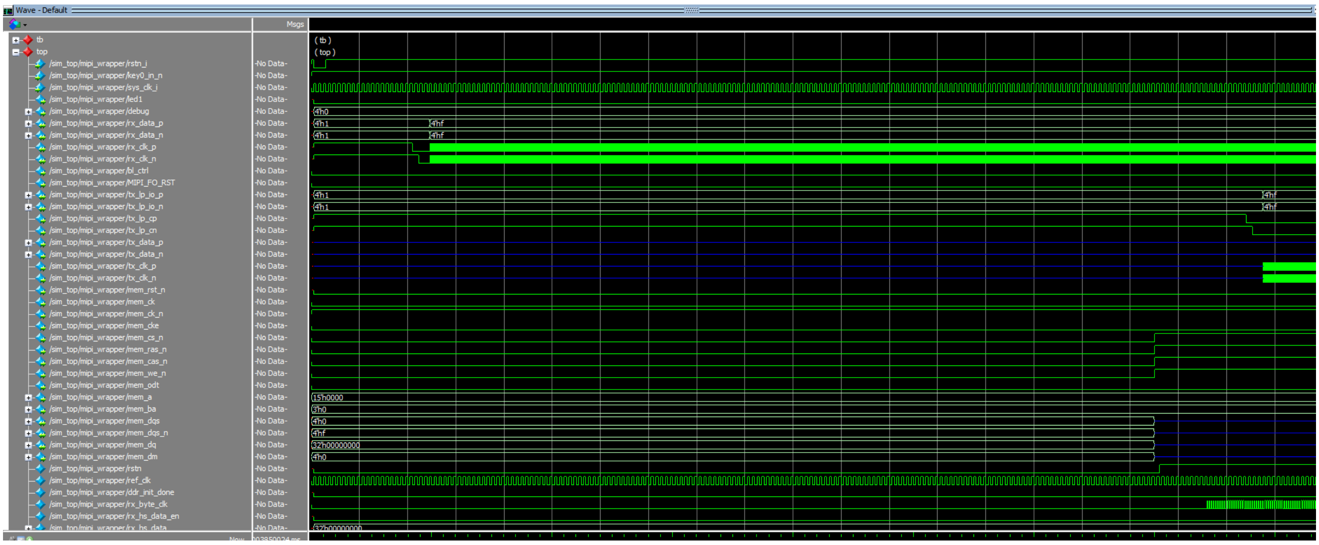


FIGURE 14 SIMULATION WAVEFORM

3.7 Reference design on-board verification

Users need to pay attention to the following before performing board verification:

(1) For mipi_rx_wrapper to work normally, first of all, it is necessary to ensure that the MIPI RX phy layer works normally. delay_clk_step or delay_data_step needs to be set to an appropriate value to ensure that the data sampling window is correct and the receiving terminal 100 ohm resistor is turned on. Secondly, it needs to be set according to the video source Mode setting DECODE_MODE parameter, "VIDEO_MODE" is used in the reference design;

(2) phy_rx needs to be reset according to the recommended reset sequence to ensure that all ISERDES work at the same time;

(3) The mipi_tx_wrapper module needs to work properly, and the mipi_tx_pll clock needs to be correctly set, as well as the timing parameters sent. It is necessary to ensure that the sent clock and data have a 90° phase difference. This can be achieved by generating two channels of the same frequency with a 90° phase difference through PLL, or through the iodelay function of the transmission;

(4) phy_tx needs to be reset according to the recommended reset sequence, and the gate function of PLL needs to be enabled;

As shown in Figure 15, turn on the power of the fixture, burn the bit stream, and the screen lights up to indicate that the board verification has passed.

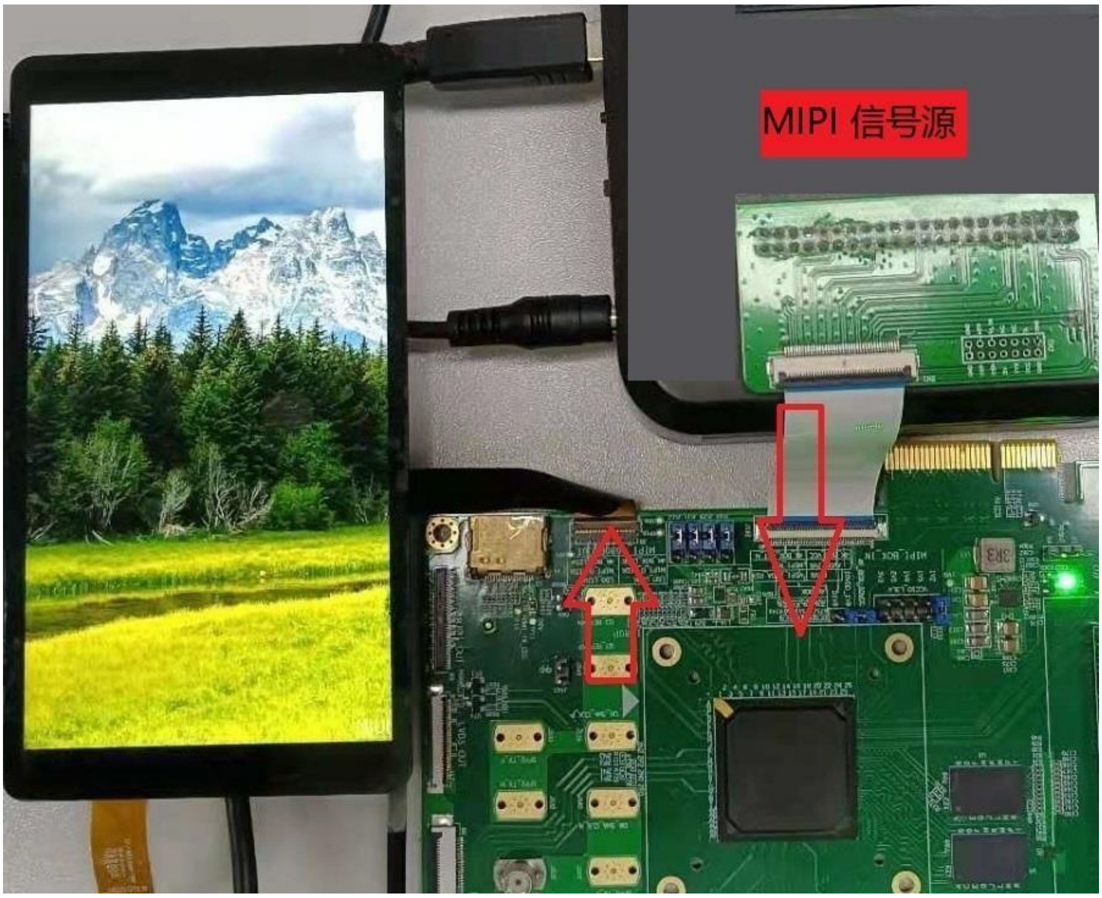


FIGURE 15 BOARD VERIFICATION